CRT Detsim to Reco: First Pass

Richie Diurba (Minnesota)

I finally got the code to work as of 10:45 am CDT

diurb001@umn.edu

Basics

When not doing first year grunt work, I have been:

- Connecting ADC hits on the CRT to channel and module locations.
- Interpreting those channel locations to real world coordinates and creating a method to stitch a 3D location from two panels in a module.
- Create a matching algorithm to make and match CRT tracks to MCC11 data.

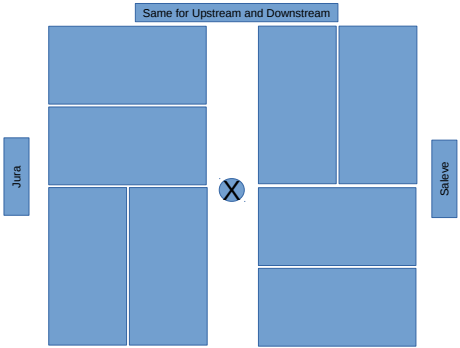
Thanks to Andrew and Arbin for laying the foundation.

Getting Hits

- Use AuxDet to get geometry of a module.
- Get geometry of a strip within the module as a channel of the hit.
- Get the center of the strip knowing beforehand that one of the XY coordinates gives reco information and the other increases scintillator target.

Basics

Find unique world coordinate for each strip and take that to be either X or Y depending on the type of strip.



CRT Orientation

Matching hits

- If two strips light up with the rough the same ADC value (only ADC values above 3000 are kept) then it is considered a hit.
- The biggest problem I had in coding this was getting rid of repetitive coordinates. This largely arose from confusion on what is a module, strip, and channel as these terms are interchangeable between the software and hardware in some cases.

Matthew Strait Aug 21

to me

Ok. So the strips are numbered 0-31 on the bottom and 32-63 on the top. By "bottom" I mean the side towards the center of each frame. Strips 0 and 63 are the ones that hang off the ends. Here's the file from Double Chooz for mapping channel (a.k.a. pixel) to strip:

Matching tracks

Use Arbin's Matching CRT module's architecture now on CRT hits.

1. Collect combinatorics of hits.
2. Compare hits to pmtracks in terms of predicted CRT hits from the pmtrack.
3. Measure displacement by looking at track points from the TPC and a slope drawn by the front and back CRTs.
4. Sort based on position of CRT track to TPC track in drift plane (x-axis).

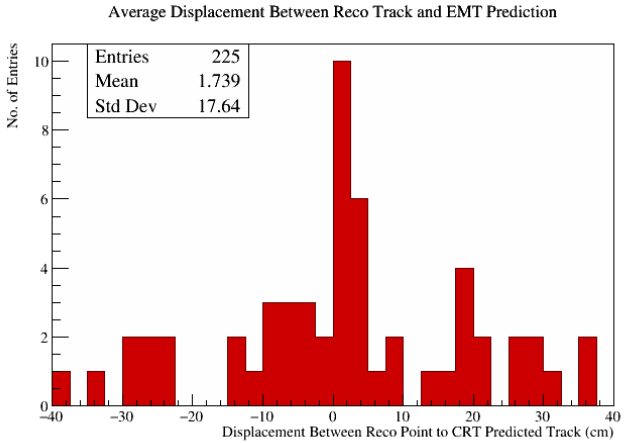
The previous Matching CRT module uses MCCheater data to construct the CRT hits so it should be used to do validation.

Issues

Where is the hit?

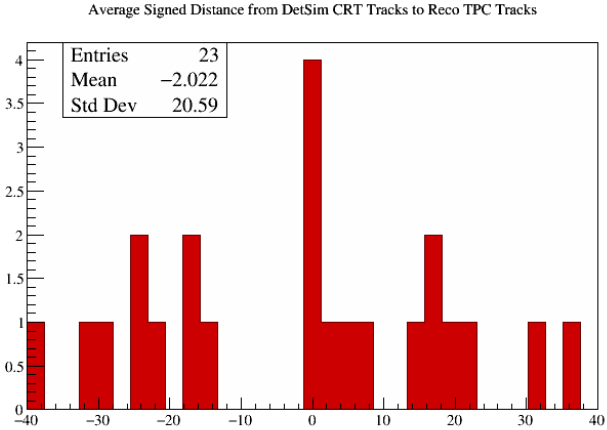
1. Collect hits
2. Take a hit (hit1) and ensure the strip measures X
3. Ensure hit2 measures y and is not a copy of hit1
4. Take hit2 as the y coordinate and average z to create 3D vector.

MCC 11 Validation



Average displacement using a toy model

MCC 11 Validation



Histogram of average displacement (cm) between CRT and TPC tracks

Moving Forward

- Optimize, debug, and cleanup code.
- Add tagging reconstruction code for the front CRT
- Add documentation

Conclusions for the Collaboration

- I need information on how to interpret ADC values given CRT performance.
- Need a bigger dataset to make conclusions (ie wait for MCC11)
- Integrate the code into Andrew's folder in Protodune/singlephase